APPENDIX B

Oracle8i Parallel Server Concepts and Administration
Release 8.1.5
A67778-01

3
Parallel Hardware Architecture

You can deploy Oracle Parallel Server (OPS) on various architectures. This chapter describes hardware implementations that accommodate the parallel server and explains their advantages and disadvantages.

Overview

This section covers the following topics:

Oracle configurations support parallel processing within a machine, between machines, and between nodes. There is no advantage to running OPS on a single node with a single instance; you would incur overhead without receiving benefits. With standard Oracle you do not have to do anything special on shared memory configurations to take advantage of some parallel processing capabilities.

Although this manual focuses on OPS on a shared nothing/shared disk architecture, the application design issues discussed in this book may also be relevant to standard Oracle systems.

Parallel Processing Hardware Implementations

We often categorize parallel processing hardware implementations according to the particular resources that are shared. This chapter describes these categories:

- Shared memory systems
- Shared disk systems
- Shared nothing systems

These implementations can also be described as "tightly coupled" or "loosely coupled", according to the way the nodes communicate.

Oracle supports *all* these implementations of parallel processing, assuming that in a shared nothing system the software enables a node to access a disk from another node. For example, the IBM SP2 features a virtual shared disk: the disk is shared through software.

Note:

Support for any given Oracle configuration is platform-dependent; check whether your platform supports your desired configuration.

Application Profiles

Online transaction processing (OLTP) applications tend to perform best on symmetric multiprocessors; they perform well on clusters and MPP systems if they can be well partitioned. Decision support (DSS) applications tend to perform well on SMPs, clusters, and massively parallel systems. Select the implementation providing the power you need for the application(s) you require.

Required Hardware and Operating System Software

Each hardware vendor implements parallel processing in its own way, but the following common elements are required for OPS:

- High Speed Interconnect
- Globally Accessible Disk or Shared Disk Subsystem

High Speed Interconnect

This is a high bandwidth, low latency communication facility among nodes for lock manager and cluster manager traffic. The interconnect can be Ethernet, FDDI, or some other proprietary interconnect method. If the primary interconnect fails, a back-up interconnect is usually available. The back-up interconnect ensures high availability, and prevents single points of failure.

Globally Accessible Disk or Shared Disk Subsystem

All nodes in loosely coupled or massively parallel systems have simultaneous access to shared disks. This gives multiple instances of Oracle8 concurrent access to the same database. These shared disk subsystems are most often implemented by way of shared SCSI or twin-tailed SCSI (common in UNIX) connections to a disk farm. On some MPP platforms, such as IBM SP, disks are associated to nodes and

a virtual shared disk software layer enables global access to all nodes.

Note:

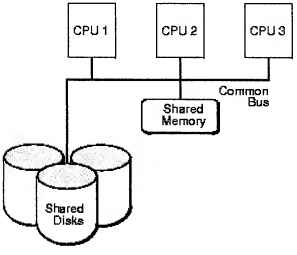
The Integrated Distributed Lock Manager (IDLM) coordinates modifications of data blocks, maintenance of cache consistency, recovery of failed nodes, transaction locks, dictionary locks, and SCN locks.

Shared Memory Systems

Tightly coupled shared memory systems, illustrated in Figure 3-1, have the following characteristics:

- Multiple CPUs share memory
- Each CPU has full access to all shared memory through a common bus
- Communication among nodes occurs by way of shared memory
- Performance is limited by memory bus bandwidth

Figure 3-1 Tightly Coupled Shared Memory System



Symmetric multiprocessor (SMP) machines are often comprised of nodes in a cluster. You can install multiple SMP nodes with OPS in a tightly coupled system where memory is shared among the multiple CPUs, and is accessible by all the CPUs through a memory bus. Examples of tightly coupled systems include the Pyramid, Sequent, and Sun SparcServer.

It does not make sense to run OPS on a single SMP machine, because the system would incur a great deal of unnecessary overhead from IDLM accesses.

Performance is potentially limited in a tightly coupled system by a number of factors. These include

various system components such as the memory bandwidth, CPU-to-CPU communication bandwidth, the memory available on the system, the I/O bandwidth, and the common bus bandwidth.

Parallel processing advantages of shared memory systems are these:

- Memory access is less expensive than inter-node communication: this means internal synchronization is faster than using the Lock Manager
- Shared memory systems are easier to administer than a cluster

A disadvantage of shared memory systems for parallel processing is:

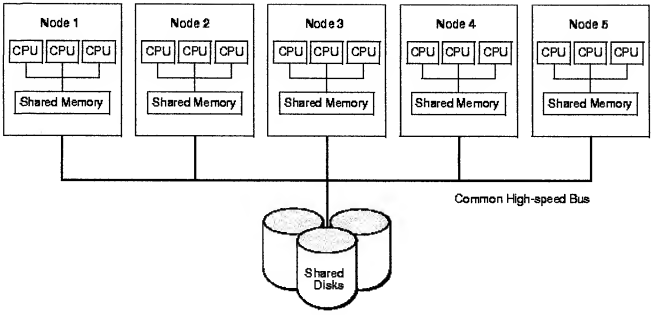
- Scalability is limited by bus bandwidth and latency, and by available memory

Shared Disk Systems

Shared disk systems are typically loosely coupled. Such systems, illustrated in Figure 3-2, have the following characteristics:

- Each node consists of one or more CPUs and associated memory
- Memory is not shared among nodes
- Communication occurs over a common high-speed bus
- Each node has access to the same disks and other resources
- A node can be an SMP if the hardware supports it
- Bandwidth of the high-speed bus limits the number of nodes (scalability) of the system

Figure 3-2 Loosely Coupled Shared Disk System



The cluster illustrated in Figure 3-2 is composed of multiple, tightly coupled nodes. The IDLM is required. Examples of loosely coupled systems are VAX clusters or Sun clusters.

Since memory is not shared among the nodes, each node has its own data cache. Cache consistency must be maintained across the nodes and a lock manager is needed to maintain the consistency. Additionally, instance locks using the IDLM on the Oracle level must be maintained to ensure all nodes in the cluster see identical data.

There is additional overhead in maintaining the locks and ensuring data cache consistency. The effect on performance is dependent on the hardware and software components, such as the high-speed bus bandwidth through which the nodes communicate, and IDLM performance.

Parallel processing advantages of shared disk systems are:

- Shared disk systems permit high availability. All data is accessible even if one node dies.
- These systems have the concept of "one database", which is an advantage over shared nothing systems.
- Shared disk systems provide for incremental growth.

Parallel processing disadvantages of shared disk systems are:

- Inter-node synchronization is required, involving IDLM overhead and greater dependency on high-speed interconnect.
- If the workload is not partitioned well, there may be high synchronization overhead.
- There is operating system overhead of running shared disk software.

Note:

Memory mapped hardware available in late 1998 will provide functionality to copy buffers directly from one user address on one node to another user address on another node.

Shared Nothing Systems

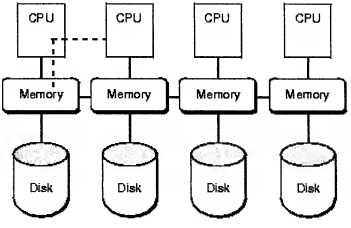
Shared nothing systems are typically loosely coupled. This section describes:

- Overview of Shared Nothing Systems
- Massively Parallel Systems
- Summary of Shared Nothing Systems

Overview of Shared Nothing Systems

In shared nothing systems, only one CPU is connected to a given disk. If a table or database is located on that disk, access depends entirely on the CPU that owns it. Figure 3-3 illustrates shared nothing systems:

Figure 3-3 Shared Nothing System



Shared nothing systems are concerned with access to disks, not access to memory. Nonetheless, adding more CPUs and disks can improve scaleup. OPS can access the disks on a shared nothing system as long as the operating system provides transparent disk access, but this access is expensive in terms of latency.

Massively Parallel Systems

Massively parallel (MPP) systems have these characteristics:

- MPP systems can range in size from only a few nodes to up to thousands of nodes
- The cost per processor may be extremely low because each node is an inexpensive processor

- Each node has associated non-shared memory
- Each node may have its own devices, but during failures other nodes can access the devices of the failed node
- Nodes are organized in a grid, mesh, or hypercube arrangement
- Oracle instances can potentially reside on any or all nodes

As mentioned, an MPP system can have as many as several thousand nodes. Each node may have its own Oracle instance with all the standard facilities of an instance. (An Oracle instance comprises the System Global Area and all the background processes.)

An MPP has access to a huge amount of real memory for all database operations (such as sorts or the buffer cache), since each node has its own associated memory. To avoid disk I/O, this advantage is important to long running queries and sorts. This is not possible for 32-bit machines which have 2GB addressing limits; total memory on MPP systems may be over 2GB. As with loosely coupled systems, cache consistency on MPPs must still be maintained across all nodes in the system. Thus, the overhead for cache management is still present. Examples of MPP systems are the nCUBE2 Scalar Supercomputer, the Unisys OPUS, Amdahl, Meiko, Pyramid, Smile, and the IBM SP.

Summary of Shared Nothing Systems

Shared nothing systems have advantages and disadvantages for parallel processing:

Advantages

- Shared nothing systems provide incremental growth
- System growth is practically unlimited
- MPPs are generally good for read-only, DSS applications
- Failure is local: if one node fails, the others stay up

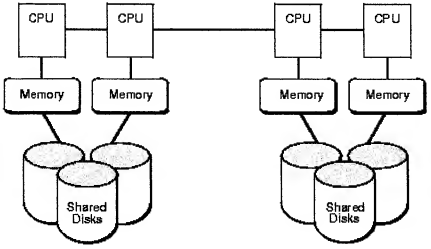
Disadvantages

- More coordination is required
- More overhead is required for processes working on a disk belonging to another node
- If there is a heavy workload of updates or inserts, as in online transaction processing systems, it may be worthwhile to consider data-dependent routing to reduce contention.

Shared Nothing /Shared Disk Combined Systems

A combined system can be very advantageous. This unites advantages of shared nothing and shared disk, while overcoming their respective limitations. Figure 3-4 illustrates a combined system:

Figure 3-4 Two Shared Disk Systems Forming a Shared Nothing System



Here, two shared disk systems are linked to form a system with the same hardware redundancies as a shared nothing system. If one CPU fails, the other CPUs can still access all disks.

Prev Next

Library Product Contents Index